Denial of Service Attacks

Tools of the Tools

By: bland_inquisitor (*bland_inquisitor@hotmail.com*) of the Digital Dawg Pound

DISCLAIMER:
All of the information contained in this article is for INFORMATIONAL PURPOSES ONLY!! I do not approve of DoS attacks used for the sake of mindless violence; I think that in this form they are the direct opposite of hacking. If you manage to use this information illegally, it's your problem not mine.

OVERVIEW:

We've all heard their names: Teardrop, Fraggle, Smurf, Bonk, and many more. DoS attacks are small, nasty, readily available, and take zero technical proficiency to use. This is a bad combination for everyone. EBay, ZDNet, CNN, and countless other systems have fallen victim to this type of criminal activity. DoS attacks cost corporations millions of dollars every year in lost productivity. In this article I hope to show the basic theories behind how a DoS attacks are possible, explain some of the generic DoS scripts out there, and show how DoS attacks have evolved into more precise and lethal tools of destruction.

TYPES OF DoS ATTACKS:

Bandwidth Consumption-

The least personal, and most easily detected, type of DoS attack is based on bandwidth consumption. How it happens, is that the attacker will eat up all the available bandwidth on the victim's system. There are 2 possible ways this can take place.

I. If the attacker has more available bandwidth than the target, he can simply flood it by being able to receive more information than he needs to send. (Ever heard the term "ping flood?")

II. Some DoS attacks, as we will see later, can be amplified by using the combined resources of another network. By doing this, an attacker can flood even the largest networks with relative ease.

If a criminal is going to DoS someone, they will most likely execute it from a system they have already "Owned," however, it is not uncommon for an attacker to deny service from their personal internet connection using a spoofed IP address. The frustrating part of this type of attack is the fact that it is based on a fundamental flaw in TCP/IP architecture: the substandard way in which systems handles SYN requests.

Resource Theft-

What if an attacker feels the need to DoS someone but doesn't have either an Owned system to send from, or a network connection capable of overpowering the target? Never fear, someone's already thought of that. A resource theft attack overutilizes access that the criminal already has to a computer to hang or crash it by using all the available memory or overtaxing the CPU. For example, an attacker could spawn multiple executions of freecell on a computer, therefore using all of the available system memory. This would result in a computer not allowing any more processes to be run, and denying service to legitimate users.

Flawed Programming-

There are other types of attacks that make full use of programming oversights. The Pentium f00f attack allows someone to crash any x86 environment by executing the bogus instruction 0xf00fc7c8 because of a flaw in Pentium microprocessor programming. We know that it is possible to execute commands in a bufferoverrun situation, and this type of attack is based on that principle. For those who may not be familiar with the term "buffer overflow," it is a condition that allows for code to be run (usually as root) by putting a greater number of characters than allowed for into a variable. The most common occurrence of this is when a program inserts data into a buffer without checking its size.

DNS Cache Poisoning-

It is also possible to alter a router so that it redirects all incoming traffic to an unintended location, either through the attacker's system, or into a non-existent one. DNS attacks or "cache poisoning," occurs when a DNS server is tricked into resolving an unintended location. An example of cache poisoning would be if someone redirected all the traffic intended to go to www.stankdawg.com to www.disney.com therefore denying service to www.stankdawg.com. Also, it is possible to redirect traffic to a non-existent network or "black hole." An example of this would be sending all incoming traffic meant for www.oldskoolphreak.com to be sent to an arbitrary address, essentially erasing www.oldskoolphreak.com from the internet. This could go undiscovered for days, until the host notices their hits went from 5000 to 0!

A LOOK AT CANNED DoS ATTACKS:

Smurf-

Smurf is a self-amplifying attack that uses directed broadcasts to crash a network. There are 3 players in this scenario: the criminal, the amplifying network, and the victim system. What happens is that an ICMP ECHO packet is spoofed to appear as though it were sent from the victim's system to the amplifying system's broadcast address. Here's where the shiznit hits the fan. Every box on the amplifying system that is configured to respond to a broadcast ping request will respond to the victim system, thereby flooding it with responses, and shutting it down. To keep your system out of the amplification business, simply disable directed broadcasting at your border router. To keep from getting "Smurfed," limit incoming ICMP and UDP at your router to only those systems that need it. If you find your system on the business end of a DoS attack, get with the amplification system, and use a tool like MCI's "dostracker" to trace the attack to its source.

Fraggle-

Fraggle, a variant of Smurf, is a DoS mechanism that uses bogus UDP packets, to port 7 (the echo port), as opposed to Smurf's ICMP. The advantage over Smurf, if you want to call it that, is that if a box on the amplification system is not configured to respond to UDP, it will send back an error message that will consume bandwidth.

DDoS ATTACKS:

In February of 2000, the long theorized DDoS attacks came. EBay fell, then CNN.com, then 5 other major systems and a myriad of minor ones came grinding to a halt. DDoS attacks require more forethought than DoS attacks, but that doesn't make them any harder to accomplish, or any less common. The difficulty is in 0wning the systems themselves!

There are 2 parts to most DDoS scripts, the client (used by the criminal), and the servers (placed on unwitting or already 0wned systems). An attacker will place the server software on as many computers as possible, making them his "zombies." Then, when the attacker feels the time is right, the zombies will execute the attack command, using their resources, and IP addresses, to shut the victim system down.

The first DDoS attack mechanism was written for *nix systems by "Mixter." The "Tribe Flood Network" offered all the standard DoS attacks, and sported a TCPbound root shell. After TFN was shown to be effective, the look-alikes hit the scene, all attempting to offer better features while simplifying the process even farther. Trinoo and Stacheldraht are 2 major players in the post-TFN market. Of the 2, Stacheldraht is the most stable and lethal of the DDoS programs. Offering ICMP, UDP, SYN, and smurf-style attacks, encrypted telnet sessions between client and server, and the ability to blind network-based intrusion detection software, Stacheldraht is the leanest, meanest way to hose a network almost anonymously.

LOCAL ATTACKS:

There are a number of local attacks, but they are not very popular. Also, they are all but outdated. These examples are more aptly defined as "exploits," but I mention them here because they can lead to a DoS situation, even though they are distant cousins. On NT 4.0, there is a way to fill %systemdrive% by exploiting disk quota functionality. In Linux kernel 2.2.0, a local attacker could use the munmap () function call used by ldd to overwrite key areas of the kernel memory, causing a kernel panic.

In closing, remember that the key word in "denial of service" is DENIAL! It's not always a matter of using brute force to shut someone down. Almost always, the most effective attacks are also the stealthiest. If you want to learn more about DoS attacks, try them out on YOUR OWN system. Learn safely, and have phun!

SHOUTS:

StankDawg, who for all the editing is hereby officially promoted to co-author, dual_parallel, and everybody at www.stankdawg.com and www.oldskoolphreak.com.